

Twelfth International Multi-Conference on Information Processing-2016 (IMCIP-2016)

A Perspective on Collaboration with Interconnects & Routing in Network on Chip Architectures

T. Praveen Blessington^{a,*}, Bhanu Bhaskara^b and Fazal Noor Basha^a

^aKL University, Vaddeswaram, India

^bCollege of Computer and Information Sciences, Majmaah University, Kingdom of Saudi Arabia

Abstract

As VLSI technology advances, the number of modules on a chip multiplies and thus the solutions for on-chip communication are evolving to support the new paradigm in inter-module communication on System on Chip (SOC). Those System on Chip, Current chip designs incorporate more complex multilayered and stack segmented interconnection buses with various routing architectures results in a Network on Chip. These, traditional solutions, which were based on a combination of shared-buses and dedicated module-to-module wires, scalability limit, and are no longer adequate for System on Chip/Network on Chip. On-chip architectures have been optimized for a non-chip environment before the multi-core challenge became the focus of processor chip architecture through the latency and the throughput. This evolution of on-chip interconnects may evoke feelings of among networking old-timers. The considerations that have driven data communication from shared buses to packet-switching networks and to routing protocols such as spatial reuse, multi-hop routing, flow and congestion control etc., will inevitably drive the challenges raised in the design of network interfaces with the segmented stack layered mechanism, and potentially managing the critical resources designed for on-chip modules.

© 2016 Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Peer-review under responsibility of organizing committee of the Organizing Committee of IMCIP-2016

Keywords: Multicores; Network-on-Chip; Router; System-on-Chip; Xpipes.

1. Introduction

Currently, the chip design is to incorporate a full-fledged network-on-a-chip (NOC) consisting of a collection of links and routers and a new set of routing protocols that govern their operation. The survey reasons for the inevitable shift to NOCs in the VLSI world, while exposing the most important requirements from the NOC¹. The aim is to expose the networking with system to the concept of network-on-chip (NOC) as a realm, within the VLSI in which the networking among the multi-cores plays a significant role in exploring the solutions such as network design, routing, and quality-of-service (QoS), unfamiliar settings under new constraints of VLSI. In order to stimulate some specific research directions, arising in each of these categories, focus is made on routing and resource allocations for the cores.

The first step was to address the low level challenges in designing on-chip interconnects in presence of deep sub-micron technologies. Due to the increased role of noise sources such as crosstalk, power-supply noise,

*Corresponding author. Tel: +91 -9730562120.

E-mail address: praveentblessington@kluniversity.in

soft errors, etc. physical link design will not suffice to provide communication reliability, and the proper course of error-control actions will have to be taken at higher levels of abstraction².

As a second step, the system level NOC design came up with an on-chip architecture, which can be used to instantiate application-specific MPSOCs consists of network building blocks/cores that can be arbitrarily tuned and composed at instantiation time on the network stack layer, and investigating how communication reliability can be traded-off with power, aware that the implementation of delays between the modules taken place³. This solution provides the flexibility at the cost and also size for an increased design complexity by placing in stacks. Two relevant features are the use of deeply pipelined switches and of link pipelining, which decouples link throughput from the worst case link delay in the design. Therefore, the operating frequencies in the order of multi-GHz range can be achieved. Xpipes is one of the most advanced NOC designs targeting heterogeneous MPSOCs with customized domain-specific communication architectures^{4,5}.

2. Routing in NOC

A router must perform two fundamental tasks: **Routing** and **Packet forwarding**. The NOC is a system of communication between the core entities segmenting into smaller modules, such that the difference between the NOC and SOC for the system, is illustrated as SOC is a single layer application centered logic device and the NOC as the multiple layered of stacks placing the applications driven on a SOC mounting with each layer so as to minimize the effectiveness of area, delay between the core applications, The NOC interconnects in different stacks, where different layers implement on different cores in the blocks of the interconnect. The power of traditional protocol stacks, such as TCP-over-IP-over-Ethernet, is such example that the information at each layer is encapsulated by the layer below it. The routing of the NOC implementation comes from the same core source and the encapsulation of such information at each layer for the protocol stack is routed with interconnects for the layers of the core data. The routing & interconnections of such modules on a generic SOC is shown in Fig. 2 below.

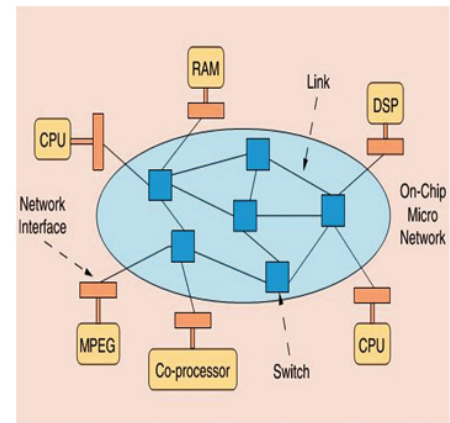


Fig. 1. An example for NOC Architecture.

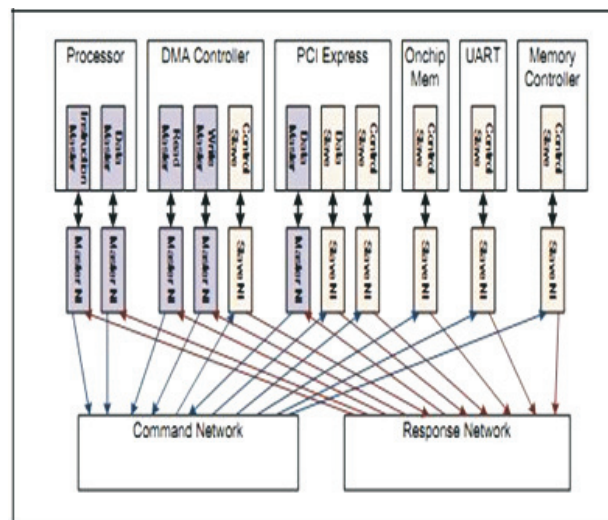


Fig. 2. Routing to Blocks in Generic SOC.

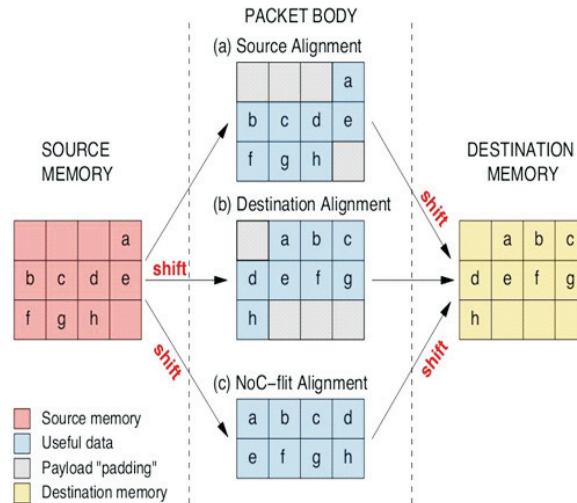


Fig. 3. NOC based Multiprocessor Context in the Stack Layers for Routing Topologies.

3. Interconnects in NOC

In NOC structures structured module can be modelled in SOC floor plan to accommodate the interconnect constraints for developing 4×4 matrix stacked layers. The possibility of these stacks makes the interconnects design with periodical measures by reducing the area to a set of synchronous module's wire segments and basic interface applications⁸. Due to this productivity increase of interconnects wires and other parametric effects yields to system level and good physical models for system performance estimations are crucial to the first success and system optimization. Now, a very successful technique for the system-level application model lied for the interconnects, and includes a large scale of integration for modules presented in the stack are implemented in physical domain aspects^{9–11} present in the interconnects design methods and performance estimation for SOC.

Based on this system design, performance estimation for SOC, the testability for various kinds of System-level decisions can thus be made with improved accuracy and support. SOC modules are best partitioned such that the higher performance of stacks layered for networking of switches can be achieved with high-speed interconnects only where necessary. One example is a SOC generic system where a memory organized hierarchically in many ways: i.e moving the L2 cache off the slow system bus and interconnecting the L2 cache for the microprocessor with bit by bit connections considerably improved by achieving higher speed and less area in connecting the variant modules of SOC in the stack shown in Fig. 3¹².

In a multiprocessor, the stack layers of a system inject a multicasting node message into a network by sending separate components of each module of the core from the source to every destination node with interconnections by routing in stacks for the design. The methodology applied in various stacks of modules are illustrated by the algorithm written in C Language. A packet one after the other it reaches one stack at its destination and we count it as one delivery at the destination node appeared. The number of deliveries at each node is achieved by an algorithm is called its throughput of the design. An unbounded number of packets is allowed to inject the adversary of the nodes at the switch, it will allow the routing algorithms to drop the packets of each switch such that a high throughput of the design can be achieved observing buffer size is as small as considerations on time and latency provided by the design allowed to intend the scheduling of data from source to destination¹⁴. The data provided by the nodes of the cores are updated with a source memory content from the source alignment applied to efficiently support the shared -data invalidation to the destination memory is detailed¹⁵ and updating on distributed nodes shown in Fig. 3.

In the Multiprocessor System-on-Chip (MPSOC) or multi-cores systems domains, parallel computing can potentially be taken into consideration on parallel programming models illustrated above Fig. 3 memory of the cores

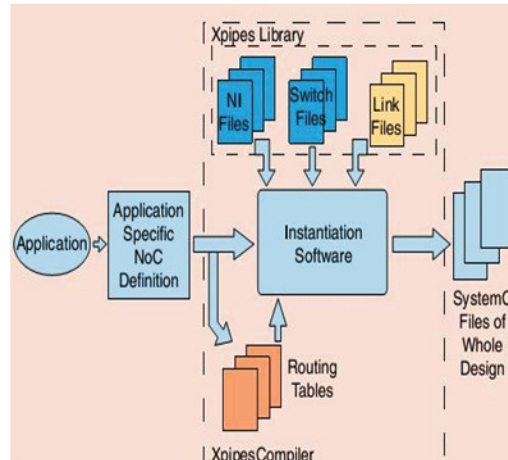


Fig. 4. Xpipes NOC Design Flow.

from source to destination. Therefore, multicast communication services, yields which should be implemented in above stack protocol layers (typically on software level) and below stack protocol layers (typically on hardware level), an important concern in a NOC based multiprocessor context leading into Multitasking communication services for its context essential for efficient implementation of memory coherency protocols which are afflicted. Furthermore, in the above stack protocol layers, multicast services are implemented as Application Programming Interface (API) routines (programming model) that can be used by users to develop parallel computing programs¹⁷.

4. The Xpipes Architecture

With a much high of application, parameters, characteristics implementation is carried over the resource constrained Multiprocessor system-on-chip (MPSOC) domain, a new paradigm is introduced intended to be a part of MPSOC is given as Xpipes Network on Chip(XNOC) architecture. The features give a high degree of parametrization and compact implementation¹⁸. XNOC is in contrast to a typical switches a fully synthesizable targeting the Chip Multiprocessor (CMP), and achieves maximum frequency that peak at around 2.5 GHz. The switch which is conceived as a macro, and the possibility of design-time for parameters such as flit width, number of Input/output ports, buffer size, and flow control is signaled using a stall & go backpressure protocol described in design flow of NOC which is shown below¹⁹. The realization for this XNOC is so specific and reliable on switching from on/off flow control protocol, and it requires two control wires: Flagging data in the forward environment and the other is signaling either a stall or go. The implementation of Stall/go with distributed buffering provides every link pipeline stage that can be designed as a two -stage First-In First-Out (FIFO).

Each switch which has carried files that are processed by the compiler needs to be linked with instantiation software i.e Noxim where the packetized communication core files in NOC makes it easier to transmit the data which is incorporated with error control information needs to be into modular segments as the application on NOC delivered in NOC design also allows error control to be implemented on segment basis⁶. Each modular core design in line with scalability concerns are designed for the complex chips. But, on chip communication differs from each module of cores which are evaluated and optimized in terms of speed, area, delay, and power consumption to improve the system-core reliability. To design a reliable NOC architectures, the issues of power consumption need to be considered, but have not been fully considered in existing works with different reliability enhancement schemes would consume different amount of hardware resources with routing algorithms where traffic delay, latency and throughput are measured. Therefore, it is necessary to devise an in-depth analysis and comprehensive experiment to explore the design space of different NOC reliability enhancement strategies¹⁵.

These strategies are further segmented into core switches and the routing for intermediate core structures are pipelined with various distributed proposals of algorithms so that the delay from the switches are going to reduce with much enhanced speed throughout the distributed level of goals. In uniform traffic, a node transmits a packet to any other node with equal probability are arranged such that the compiler can do the tasks easily providing a design where the set of nodes are to be less weighted. The interconnections of 4×4 mesh topology is derived and shown in Fig. 5.

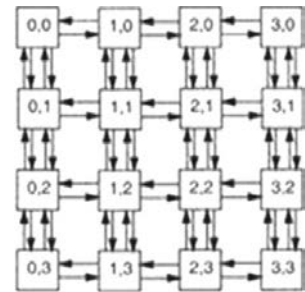


Fig. 5. 4×4 Routing Switch.

A comparative frame work of different NOC routing topologies would offer a cost-benefit comparison framework of different NOC reliability schemes so that the proposed methods can be compared with existing methods under the same set of objective criteria, and also to provide a guideline on how Error Correction Code (ECC) measures should be chosen under different design constraints in a 4×4 matrix switch. Once the packet reaches the block, it is forwarded through the appropriate data line card. As routers keep forwarding the packets from various blocks, there can be an instance of different accessible network interfaces need to be forwarded to the same network interface simultaneously. The buffers are stranded such that the receiving data lines from various sources cores to destination resources data is accessible via interconnects¹⁷ are detailed in the design.

5. Implementation & Results

The proposed XPipes NOC architectures of two bidirectional channels to NOC were used such that in each pair of neighboring routers, the performance of various core block is routed to inspect the performance trend on different buffer configurations and switching strategies were implemented on both the wormhole and virtual-channel flow-control based on various router architectures is explained. The virtual-channel flow-control, were configured with different numbers of virtual-channels provides two, three, and the variation of average packet latency for 64 Bit of packets, 32 bit of throughput) with linked bandwidth frequency of 2.5 GHz. NOCs, as synthesized by Xpipes NOC Noxim Compiler, have lower packet latencies, as their average number of switches is lower with a 10% latency is achieved and the throughput is moderate between the channels exits. Moreover, the latency increases more rapidly with the mesh NOC as the link bandwidth decreases. The comparative analysis is tabled below.

The detailed simulation environments are described below and the router architectures used for comparison are listed in Tables 1 and 2. In Table 1, The latency of various routing algorithms are listed with their Avg. packet of channel bandwidths, which was divided into four 8-flit buffer queues used as virtual-channels. The data transmitted is done for each pair of neighboring routers with one input channel and one output channel of various topologies are introduced. Besides, a typical unidirectional NOC architecture with wormhole flow-control which occupied one 32-flit buffer queues in each direction has moreover equipped with four unidirectional links between adjacent router pairs was carried & implemented to evaluate the effects of doubling the inter-router communication bandwidth for the core data.

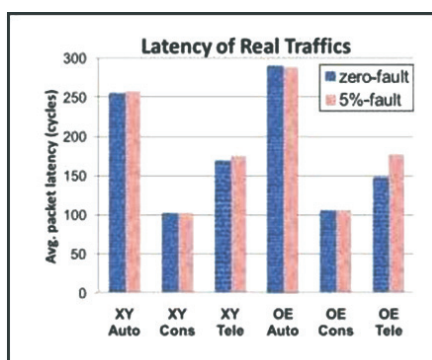


Table 1. Latency of Real Traffic.

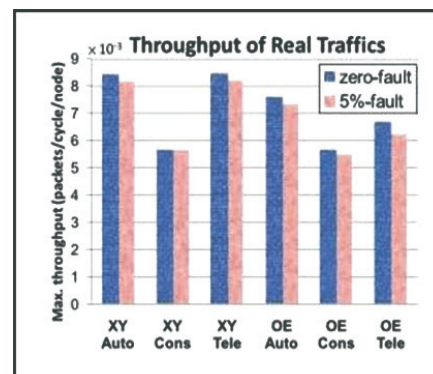


Table 2. Throughput of Real Traffic.

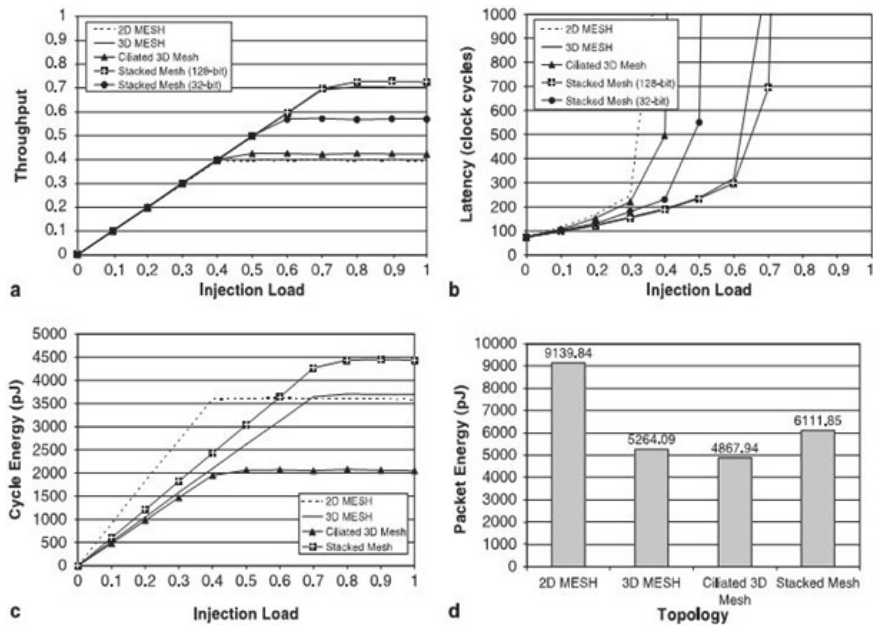


Table 3. Latency vs Throughput for Various Routing Topologies in NOC.

Also, Xpipes NOCs have better link utilization compared to traditional NOCs of around 1.5 times the link utilization of a mesh routing topology implemented. It should be observed that area, power and performance optimizations by means of NOCs turn out to be half the cores needed to communicate with more than a single core. This motivates a configuration of the Xpipes NOC, having less than half the number of switches than the mesh NOC. In this way, the xpipes NOC consumes about 5.7 times less area and 2.7 times less power than the corresponding mesh traditional NOC. The various graphs for bandwidth of different routing topologies are discussed in comparison with traditional NOC is tabled below.

6. Conclusions and Future Work

The various challenges for NOC from shared buses to packet-switching networks for routing protocols were illustrated by the Mesh routing algorithm will inevitably drive the challenges raised in the design of network interfaces from the traditional NOCs. With the segmented stack layered mechanism, and potentially managing the critical resources designed for on-chip modules, where XY routing appears as the best one in most situations, for medium to large NoCs. The second identity is to determine the best packet size and the total time to deliver the total load for various protocols that are being illustrated with the medium sized packets are the best choice, due to the network buffering capacity in compared with small packets under utilizing the stack network and to increase segmentation while the large packets lead to network congestion and buffer saturation.

It is possible to say that deeper traffic analyses are needed, using e.g. real traffic load distributions. Besides it is also important to consider the usage of NoCs with virtual channels, which modify blocking conditions and thus change traffic characteristics. Combining the results provided here and extensions to deal with traffic in virtual channels NoCs, it is possible to safely address the problem of designing NoCs with controlled topologies.

References

- [1] L. Benini and G. De Micheli, Networks on Chips: A New SoC Paradigm, *IEEE Computer*, vol. 35(1), pp. 70–78, (2002).
- [2] K. Bernstein, P. Andry, J. Cann, P. Emma, D. Greenberg, W. Haensch, M. Ignatowski, S. Koester, J. Magerlein, R. Puri and A. Young, Interconnects in the Third Dimension: Design Challenges for 3D ICs, *Proc. DAC*, pp. 562, (2007).

- [3] J. H. Bahn, S. E. Lee and N. Bagherzadeh, On Design and Analysis of a Feasible Network-On-Chip (NoC) Architecture, *Fourth International Conference on Information Technology, 2007, ITNG'07*, pp. 1033–1038, (2007).
- [4] K. Chang, J. Shen and T. Chen, Evaluation and Design Trade-Offs Between Circuit-Switched and Packet-Switched NOCs for Application-Specific SOCs, In *DAC'06: Proceedings of the 43rd Annual Conference on Design Automation*, pp. 143–148, (2006).
- [5] W. J. Dally and B. Towles, Route Packets, not Wires: On-Chip Interconnection Networks, *Design Automation Conference, Proceedings*, (2001).
- [6] W. J. Dally and B. Towles, Principles and Practices of Interconnection Networks, *Morgan Kaufmann*, (2004).
- [7] S. Stergiou and G. De Micheli, Xpipes Lite: A Synthesis Oriented Design Library for Networks on Chips, *Design, Automation and Test in Europe, Proceedings*, (2005).
- [8] G. Martin and H. Chang, System on Chip Design, In *9th International Symposium on Integrated Circuits, Devices & Systems (ISIC'01)*, Tutorial 2, (2011).
- [9] S. Kumar, et al., A Network on Chip Architecture and Design Methodology, In *IEEE Computer Society Annual Symposium on VLSI, (ISVLSI'02)*, pp. 105–112, April (2011).
- [10] K. Chang, J. Shen and T. Chen, Evaluation and Design Trade-Offs Between Circuit-Switched and Packet-Switched NOCs for Application-Specific SOCs, In *DAC'06: Proceedings of the 43rd Annual Conference on Design Automation*, pp. 143–148, (2006).
- [11] Pan Hao, Comparison of 2D MESH Routing Algorithm in NOC, *2011 IEEE 9th International Conference on ASIC (ASICON)*, pp. 25–28, October (2011).
- [12] T. Nesson and S. L. Johnsson, ROMM Routing on Mesh and Torus Networks, In *Proc. ACM Symposim Parallel Algorithms and Architectures*, pp. 275–287, ACM Press, (2010).
- [13] T. Praveen Blessington and Dr. B. Bhanu Murthy, Optimal Implementation of UART-SPI Interface in SoC Devices, *2012 International Conference on Circuits and Systems (ICDCS)*, ISBN: 978-1-4577-1545-7, pp. 15–16, March (2012).
- [14] Z. Zhen, A. Greiner and S. Taktak, A Reconfigurable Routing Algorithm for a Fault-tolerant 2D-Mesh Network-On-Chip, *45th Design Automation Conference Proceedings*, pp. 441–446, (2008).
- [15] T. Ahonen et al., Topology Optimization for Application Specific Networks on Chip, *Proc. SLIP 04*.
- [16] V. F. Pavlidis and E. G. Friedman, 3-D Topologies for Networks-on-Chip, *IEEE TVLSI*, October (2007).
- [17] B. Feero and P. P. Pande, Performance Evaluation for Three-Dimensional Networks-on-Chip, *Proc. ISVLSI*, (2007).
- [18] C. Addo-Quaye, Thermal-Aware Mapping and Placement for 3-D NoC Designs, *Proc. SOCC*, (2005).
- [19] J. Kim et al., A Novel Dimensionally-Decomposed Router for on-chip Communication in 3d Architectures, *ISCA*, (2007).
- [20] F. Li et al., Design and Management of 3D Chip Multiprocessors Using Network-in-Memory, *ISCA*, pp. 130–141, (2006).
- [21] D. Park et al., MIRA: A Multi-Layered On-Chip Interconnect Router Architecture, *Proc. ISCA*, (2008).
- [22] S. Murali et al., Synthesis of Networks on Chips for 3D Systems on Chips, *ASPDAC*, (2009).